

**REMARKS**

**I. Summary of the Office Action and this Reply**

Claims 1-14 and 17-20 are pending in the application; claims 15 and 16 have been withdrawn from consideration as drawn to a non-elected invention. The undersigned hereby affirms the provisional election of claims 1-14 and 17-20, without traverse.

The Examiner has rejected claims 1-3, 7, 13-14 and 17 under 35 U.S.C. §102(e), asserting anticipation by U.S. Patent No. 6,662,159 to Chao et al. ("Chao"). The Examiner has rejected claims 4-6 and 8 under 35 U.S.C. §103(a), asserting obviousness based on Chao in view of U.S. Patent No. 6,675,382 to Foster ("Foster"). The Examiner has rejected claims 9-12 and 18-20 under 35 U.S.C. §103(a), asserting obviousness based on Chao in view of U.S. Patent No. 5,729,761 to Murata et al. ("Murata").

**II. Discussion of Cited Art**

**U.S. Patent No. 6,622,159 to Chao et al.**

Chao discloses a method and apparatus for automatically restarting a remote procedure call (RPC) server without losing client RPC calls. Generally, client calls stop being received after shutting down a current server version. Requests cannot be processed immediately after a current server version stops listening for client requests and closes its socket. Instead, TCP/IP architecture requires waiting for a period of time before allowing a socket to close and be reused. Col. 1, lines 27-39. Accordingly, any client calls received after a first server version is shut down, and before a second server version is established, are lost, which results in a disruption

of service.

In one aspect, Chao discloses providing a static port with a socket bound thereto, such that the server is operable to leave the socket open when a current server version is switched to a new server version so that the server is able to continue to receive client calls received through the socket during switching. A buffer is provided for queueing client calls received during switching from the current server to the new server. The server is adapted to have the new server version process any queued client calls received during switching after switching to the new server version is complete. Col. 2, lines 23-39; col. 3, lines 36-56.

### **III. Response to 102 Rejections**

The Examiner has rejected claims 1-3, 7, 13-14 and 17 under 35 U.S.C. § 102(e), asserting anticipation by Chao.

A rejection under 35 U.S.C. § 102 is proper only if each and every element of the claim is found in a single prior art reference. MPEP § 2131.

#### **Claims 1-3, 7, 13 and 14**

Claim 1 is directed to a method for upgrading at least one of a plurality of computer programs stored on an application server in a distributed computing environment. The method involves (1) preventing the application server from servicing client's requests for a first computer program, while simultaneously (2) permitting the application server to service client requests for other computer programs retrievable from the application server. Accordingly, an entire server is not taken offline when a single application supported by the server needs to be

upgraded. Instead, the server continues to service client requests for other computer programs while first computer program is being upgraded.

For example, consider a simplified example in which three software application programs (A, B and C) are stored on a single server computer. Typically, as in Chao, when a certain software application (e.g., A) is to be upgraded, the entire server is taken offline, and thus the server cannot service any requests for any programs, such as A, B, or C. However, in accordance with the present invention, an entire server is not taken offline. Instead, requests for the upgraded application program (A) are refused while requests for application programs B and C are still accepted and serviced by the server during the upgrading of application A.

Chao requires, in contrast to the claimed invention, that an entire server be taken offline in that it is the server software itself that is upgraded. No client calls for any programs are serviced by the server while the server is offline. Instead, Chao discloses buffering all client calls while the server is offline, and such client calls are not serviced during that time, but are instead serviced later when the server is brought back up.

In contrast to Chao, the claimed invention allows the server to process client calls for programs B and C while program A is being upgraded. Accordingly, there is no need to buffer any client calls for programs B and C while program A is being upgraded. During the upgrade of program A, any client calls for program A can be serviced by another server computer in a logical application server cluster. Page 5, line 22 - page 6, line 11. For at least this reason, claim 1 is patentable.

Further, Chao neither teaches nor suggests preventing the server from servicing requests on a per application basis. In other words, in contrast to the

present invention, Chao neither teaches or suggests preventing the server from servicing requests for a certain program, while simultaneously permitting the server to service requests for other computer programs retrievable from the application server. Page 8, lines 4-15. For this additional reason, claim 1 is patentable.

Further, claim 2 recites preventing the application server from receiving any new requests for the to-be-upgraded computer program, waiting until all of the application server's current requests for the computer program have ended; acknowledging completion of upgrading of the computer program, and permitting the application server to receive any new requests for the computer program. Contrary to the Examiner's assertion in paragraph 2 of the Action, Chao provides no teaching or suggestion of any such handling of requests or calls on a per application basis. Chao merely discloses taking a server offline, during which time the server is unavailable to service any calls for any programs, and the restarting the server. Any incoming calls while the server is offline and buffered for later processing. Further, Chao provides no teaching or suggestion whatsoever of waiting until all of the application server's current requests for the computer program have ended, or otherwise monitoring client sessions on a per application basis. For these additional reasons, claim 2 is patentable.

Claim 3 further recites that preventing the application server from receiving new requests involves sending a signal to a router to instruct the router to stop routing requests for the to-be-upgraded computer program to the server. The Examiner states that "Chao's system buffers any received client calls during switching until the upgrade is completed." While this is true, the buffering occurs at the server 110. Col. 5, lines 19-22. Accordingly, the router never stops routing of

client requests to the server, and the router is never instructed to stop routing such requests to the server. Accordingly, Chao neither teaches nor suggests instructing the router to stop routing requests to the server. Claim 3 is patentable for this additional reason.

In addition, claim 7 recites that a signal is sent to a router to instruct the router to begin routing requests for the to-be-upgraded computer program to the application server, e.g. after the program has been upgraded. The Examiner states that "Chao's system processes any buffered client calls received during switching to be service by the server." This statement is unclear. It is, however, clear that Chao neither teaches nor suggests that a router stops, or starts again, routing requests to the server. Instead, in Chao, any router continuously routes requests to the server, and the server buffers any requests received while the server version is being switched. See discussion of claim 3, above.

Claims 13 and 14 depend from claim 2, and are likewise patentable. In addition, claim 13 recites repeating steps (a) through (d) for a next application server storing the same computer program, such that a software upgrade continues across multiple servers that may service a request for the computer program. Contrary to the Examiner's assertion, this is neither taught nor suggested by Chao. The Examiner is reminded of the requirement of 37 CFR §1.104(c)(2) to identify the particular part of a reference that is relied upon, and is invited to identify with specificity any portion of Chao believed to disclose such repetition of such steps.

For at least these reasons, reconsideration and withdrawal of the rejections of claims 1-3, 7, 13 and 14 are requested respectfully.

**Claim 17**

Independent claim 17 is directed to a method for upgrading one of a plurality of computer programs stored on an application server in a distributed computing environment. The method involves instructing a router to stop routing requests for the computer program to the application server. As discussed above with respect to claim 3, there is no stopping of routing of client requests to the server in Chao, and the router is not so instructed. Instead, Chao discloses that the router continuously routes requests to the server, and the server buffers the requests. Col. 5, lines 19-22. Accordingly, Chao neither teaches nor suggests instructing the router to stop routing requests to the server. Additionally, Chao neither teaches nor suggests instructing the router to later being routing requests to server again, as discussed above with respect to claim 7.

Further, there is no teaching or suggestion in Chao of monitoring the server's active sessions, and waiting until the server is no longer supporting a current client request for the computer program.

For at least these reasons, reconsideration and withdrawal of the rejection of claim 17 are requested respectfully.

**IV. Response to 103 Rejections**

The Examiner has rejected claims 4-6, 8, 9-12 and 18-20 under 35 U.S.C. § 103(a), asserting obviousness based on Chao in view of Foster or Murata.

However, it is noted that the subject matter of the Chao patent, which the Examiner has identified as prior art under 102(e), and the presently claimed invention were, at the time the invention was made, owned or subject to an obligation of assignment to the same entity, namely, the assignee hereof.

Therefore, the Chao patent cannot form the basis of a 103 rejection under 35 U.S.C. § 103(c), as the instant application was filed after November 29, 1999. Accordingly, reconsideration and withdrawal of the rejections of claims 4-6, 8, 9-12 and 18-20 are requested respectfully.


In any event, claims 4-6, 8, 9-12 and 18-20 depend from allowable claims, as discussed above, and are therefore allowable.

### **CONCLUSION**

In view of the foregoing amendments and remarks, Applicants believe claims 1-14 and 17-20 to be patentable and the application in condition for allowance. Applicants respectfully request issuance of a Notice of Allowance. If any issues remain, the undersigned requests a telephone interview prior to the issuance of an action.

Respectfully submitted,

Dated: April 29, 2004

  
Gregory S. Bernabeo  
Registration No. 44,032

Synnestvedt & Lechner LLP  
2600 Aramark Tower  
1101 Market Street  
Philadelphia, PA 19107-2950  
Telephone: 215-923-4466  
Facsimile: 215-923-2189